SpyLoc: A Light Weight Localization System for Smartphones

Mostafa Uddin
Old Dominion University, USA
muddin@cs.odu.edu

Tamer Nadeem
Old Dominion University, USA
nadeem@cs.odu.edu

ABSTRACT

In this paper, we propose and addresses the challenge of designing a light-weight high-accuracy indoor/outdoor localization system (*SpyLoc*) for off-the-shelf smartphones. In SpyLoc, we want to leverages both the acoustic interface (microphone/speaker) and the Wi-Fi interface at the kernel-level of the smartphones as well as the inertial sensors in the smartphones to achieve high localization accuracy. In the ranging-based approach, we utilizes the RF-Beep ranging scheme [**?**], and in dead reckoning-based approach, we fuses the inertial sensors of the smartphone to estimate the direction and the distance of user's movements.

In RF-Beep [**?**], we develope a ranging scheme that utilizes the Time-Difference-of-Arrival (TDoA) between the acoustic and the radio-frequency (RF) signal. The RF-Beep basically leverages the slow propagation speed of the acoustic signal with respect to the RF signal to estimate the relative range. The well known acoustic range based localization scheme, Cricket [**?**] also utilizes the same concept of using the difference in arrival times of concurrent transmissions of radio and ultrasound signals at the target device to infer the distance. Unlike the Cricket, which was designed with special hardware, our localization scheme is applicable to the smartphones. In RF-Beep, we address the different challenges of implementing such ranging scheme in smartphones by leveraging the existing functionalities of the audio driver and the WiFi driver. Further details on RF-Beep scheme could be found in [**?**]. The ranging based localization scheme typically requires at least three reference points (e.g., anchor points), in order to calculate the location. However all these three reference points must be in Line of Sight(LoS) to the target device. In *SpyLoc* localization system, we use the combination of both ranging-based and dead reckoning approaches to reduce the constraint of having three LoS anchor/reference points all the time.

The basic idea of the SpyLoc is to leverage the benefits of both the dead-reckoning and the ranging scheme to build a practical localization system. Given the high errors of the inertial sensors, SpyLoc uses a novel ranging scheme based on both the acoustic and WiFi interfaces to mitigate this error in order to improve the localization accuracy. Unlike the ranging-based or RF-based localization schemes that require multiple reference points (e.g., achor points), using the dead reckoning in SpyLoc reduces the required number of reference points to only one reference to locate and track users accurately. This low dependency on ranging scheme make SpyLoc practically applicable to high mobility environment. Finally, In the SpyLoc system, user's device (i.e. smartphone) works autonomously to determine its location. This system does not require any coordination from the nearby smartphones or from a central controller. Furthermore, this localization system does not require user's device to transmit any acoustic signal or RF messages to nearby smartphones. Therefore increasing the number of user's devices have no impact on the complexity of the SpyLoc system, which make SpyLoc a light-weight localization system. In addition, such characteristics make the localization system privacy preserving and energy efficient for the user's smartphone.

MobiCom'13, September 30–October 4, Miami, FL, USA.

ACM 978-1-4503-1999-7/13/09.
http://dx.doi.org/10.1145/2500423.2504581.

Categories and Subject Descriptors

H.4 [**INFORMATION SYSTEMS APPLICATIONS**]: Miscellaneous

Keywords

Localization, Mobility, Smartphones, Range

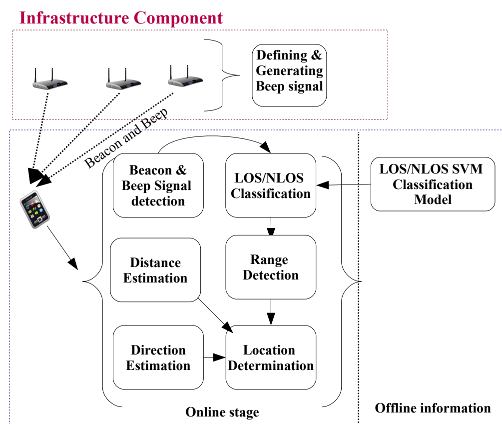


Figure 1: Architecture overview of the different components of SpyLoc localization system

1. SPYLOC LOCALIZATION SYSTEM

In this sections, we introduce the SpyLoc system architecture and the challenges. Figure 1 shows the different modules of SpyLoc system. Similar to many localization systems, SpyLoc consists of two main components: infrastructure component that runs on infrastructure hardware, and client component that runs on the user's device (i.e. smartphone). In this paper, we refer to an infrastructure device running the SpyLoc infrastructure component by a *beacon device*. The beacon device periodically broadcasts a RF message (i.e. Wi-Fi beacon frame), which we refer to as a *beacon*. In addition, the beacon device also generates an acoustic signal, *beep* following each broadcasted beacon message. In a practical scenario, a beacon device could be an Access Point (AP) with additional acoustic interface (i.e. speaker, mic and sound driver). A user's smartphone running SpyLoc client component (e.g., SpyLoc application) will capture the beacon messages and

the corresponding beep signals from the surrounding beacon devices. Using the captured beacon messages and the corresponding beep signals in addition to the inertial sensors in smartphone, SpyLoc application will infer the user's location.

In a typical usage scenario, while the user runs the SpyLoc application in the smartphone, it will initially collect the beep signals from the surrounding beacon devices along their corresponding beacon messages. After collecting the *beep* signal and the *beacon* message, the application determines the relative ranges between the user's smartphone and the corresponding beacon device. Then, the application combines three estimated ranges from three different beacon devices to estimate the user's initial location. After fixing the user's initial location, later on SpyLoc uses the inertial sensors to track the distance and the direction of the user's locations. However, In indoor environments, using the inertial sensor of the smartphone to detect the distance and the direction of the user's movement is extremely challenging due to surrounding noises from the ferromagnetic devices. Therefore, In SpyLoc we utilize the range estimation to calibrate and calculate the accurate location of the user. However, In SpyLoc, our objective is to just limit the range calculation from one beacon device.

In summary, SpyLoc localization system possess the following interesting features:

- SpyLoc exploits the multiple interfaces in the smartphones by utilizing both the RF interface and acoustic interface to design high-accurate localization scheme.
- SpyLoc does not require any central controller or backend server to coordinate between neighboring devices. In this system, user's smartphone calculate the location locally and no collaboration with neighboring devices is required. In addition, The user's application does not require to share any acoustic or WiFi information with any nearby device or access point. This enables to preserve and protect the user security and privacy.
- The SpyLoc application running by the users does not require any customized hardware. This enables any off-the-shelf smartphone to adopt and use the developed application.
- SpyLoc only needs one ranging estimation from the beacon device after knowing the user's initial position. In addition, such ranging requirement is only required to calibrate the estimated location by the dead reckoning approach. This enables SpyLoc to be light-weight and practically applicable during user's mobility.
- SpyLoc only requires one way transmission; the transmission of the beacon messages and the beep signals by the beacon devices. Furthermore, estimating one range from one beacon device eliminates the need to switch the Wi-Fi channel to receive multiple beacon messages from the multiple beacon devices. Therefor, the development of the SpyLoc is energy efficient application for user's smartphones.

In the following subsection, we provide a brief overview of the infrastructure component and the client component.

1.1 Infrastructure Component

The beacon device (e.g. Wi-Fi AP), which runs the infrastructure component of the SpyLoc system, periodically generates a RF beacon message followed by a beep signal. A single frequency sinusoidal acoustic signal defines the basic sound that we refer to as *tone* in the paper. A mixture of such tones (i.e., set of frequencies) defines the beep signal. Typical human hearing perception diminishes after 18kHz. Therefore, we utilize the 18kHz-21kHz audio frequency range, which is perceptible to the most of the off-the-shelf smartphones [**?**]. From experiments, we found that if the frequency space between two adjacent tone is 250Hz, then it is sufficient to avoid the interference and detect the tones on the client side. Therefore, we select 10 tones(i.e frequencies) $(f_1, f_2, ..., f_{10})$ from 18kHz-21kHz audio range, we can have up to 2^{10} unique beep signals. One of the major challenge in the infrastructure component is, *How to uniquely and autonomously define the tones of the beep signal for each beacon device?*

In selecting the tones we utilize the last ten bits of the MAC address from the Wi-Fi Interface card of the beacon device. In this last ten bit sequence, each bit position correspond to one tone among the 10 tones, $\{f_1, f_2, ..., f_{10}\}$ (i.e. the $0th$ bit map to f_1, $1th$ bit map to f_2, and so on). A value of 1 in one bit position indicates that corresponding tone will exists in the beep signal and vice versa for the value of '0'. For example, if the MAC address of a beacon device is $c4 : 2c : 03 : 3a : 2c : a1$, that has last ten bits as 0010100001, then the selected tones of the beep signal for that beacon device would be $\{f_1, f_6, f_8\}$. Selecting a tone is such way has almost 1% ($= (10*100)/2^{10}$) chance of having same tones for two different beacon device, if we consider 10 beacon devices in one place within each other proximity. Note that, at different public places we didn't found any cases where two Wi-Fi AP have same last 10 bit MAC address in one place.

In addition with the above designing of the *beep* signal, designing the RF message for each beacon device is challenging. In designing the infrastructure component, we leverage the existing WLAN setup, so that we can reduces the barrier of deploying our localization system. Therefore, we use the existing WLAN to generate the RF beacon message without imposing any overhead on the WLAN traffic. For example, we use the Wi-Fi beacon frame as our beacon message. In addition, we add the location information of the AP in the payload of the the Wi-Fi beacon frame, so that the client application can calculate the location locally. Furthermore, In the scenario of multiple APs, we also need to address the challenge of mapping each *beep* signal with the corresponding RF *beacon message* from a AP. As defining the beep signal in a unique way for each beacon device, help us to reduce the ambiguity of mapping the beep signal to the correct beacon message. In the implementation, we use a timeout of 100ms of detecting the beep signal after receiving the corresponding beacon message, which is also the normal periodicity of the beacon message (i.e. beacon period).

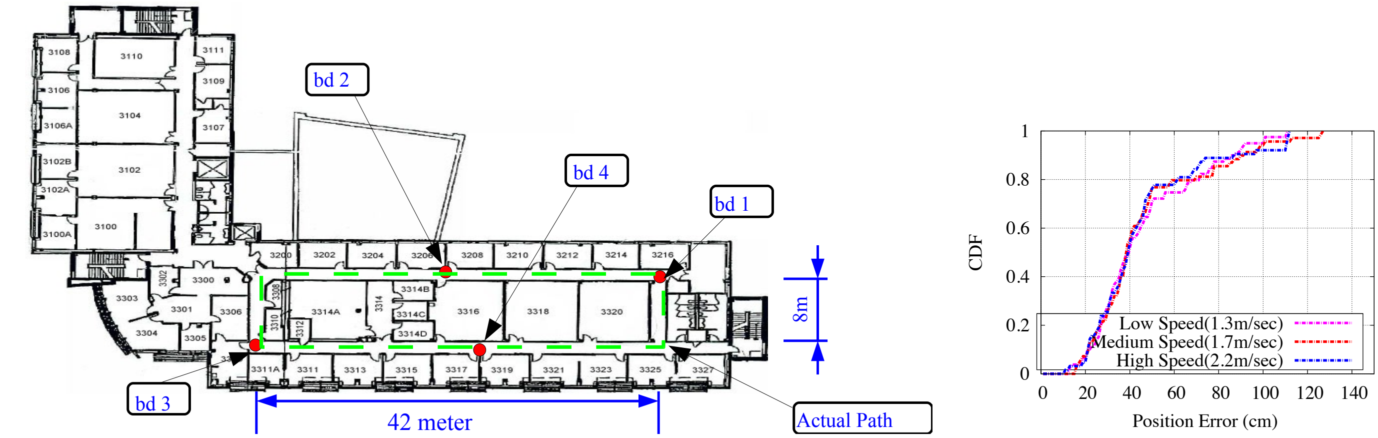
1.2 Client Component

In SpyLoc, client component consists of following modules:

Distance Estimation module: This module has two submodules i) *Step Detector*, and ii) *Personalized Step Model*. The step detector uses the commonly used accelerometer and the gyroscope sensors of the smartphone to detect and track the user steps. Typically, these sensor reading shows a noticeable periodic pattern for human steps, unless the smartphone is not carried by the user. However, the duration of each periodic pattern varies dynamically with the change of human walking speed. Numerous, paper address this challenge by building a standard model to consider unique fixed length step duration for each user [**?**]. The dynamic nature of human walking speed is quite natural in indoor environment. Therefore, using a fixed step duration for each user might create error in each step, which will accumulate over the time. Therefore, In SpyLoc we leverages our own customized Dynamic Time Wrapping (DTW) algorithm propose a robust and online step duration detection technique that address the dynamic nature of human walking speed. In the implementation, we use a fixed step length and our own DTW algorithm to calculate the final distance traveled by the user.

Direction Estimation: This module utilizes the geomagnetic and gyroscope sensors to detect the direction of the user. In indoor environments, using the inertial sensor of the smartphone to detect the direction of the user's movement is extremely critical due to surrounding noises from the ferromagnetic devices. Therefore in current implementation, we use the function *getRotationMatri* provided by the android API to collect more cleaner ration sensor reading.

Location Determination module: The location determination module is responsible to estimate the user's final location based on both the range estimation and the dead reckoning approach using inertial sensors. If the user is in the LoS of at least three beacon devices, location determination module applies the triangulation technique to estimate the user's actual location. However, in scenarios where the users is moving, (e.g. walking or running) such triangulation technique is not always practically applicable due to time limitation of computations. In addition, existence of three beacon device (e.g WiFi AP) in LoS is not always a practical scenario. Therefore, our location determination module utilizes the inertial sensors to predict the user's location.



9(a) 9(b)

Figure 2: (a) Indoor experiment setup at department building, following the green line as a route. (b) CDF of the estimation error for following the route in 2(a) at different speed.

However, such technique is not highly reliable in predicting the location. Since localization error in dead reckoning scheme get cumulated over time. Thus, instead of using three ranges from three beacon devices, our location determination module uses one range estimation to calibrate the estimated location from the inertial sensors. In situation where no LoS beacon device exists, location determination module relies on the inertial sensors to predicts the location.

The variation in direction estimation due to the noise, we assume the following gaussian model for the rotation sensor reading θ,

$$\theta = \mathcal{N}(\theta_\mu, \theta_\sigma), \tag{1}$$

where θ_μ is the mean and θ_σ is the variation. Therefore, lets assume the direction/distance estimation module provides possible set of locations,

$$L = \{l_1, l_2, ..., l_i, ..., l_N\}$$
$$l_i \equiv (x_i, y_i) \; i = 1, 2, .., N \tag{2}$$

Now In SpyLoc, with a single LoS, we use the following calculations to infer the user's final location by selecting the location point that is closest to the range r from the only beacon device.

$$d_{min} = \min_{l_i} |r - \sqrt{(a - x_i)^2 + (b - y_i)^2}| \tag{3}$$

where a, b are the center coordinate of the only LoS beacon device.

2. EVALUATION

Figure 2(a) shows the location of the beacon devices and the actual followed route(green line). Note that, during this experiment we use one range estimation and the inertial sensors to calibrate user's final location.The plot 2(b) shows the CDF of the estimation while we follow the route in 2(a) at different speed. In this experiment, we found estimation error less then 90cm for 90% of the times for different speed. These results proofs the feasibility and the efficiency of our SpyLoc system under different mobility condition of the user.

3. CHALLENGES AND FUTURE WORK

In future work we like to address the following challenges in details:

- The commonly used human step length model [**?**]infer the user's step length from the time duration of a steps.

$$s = \frac{a}{f} + b, \tag{4}$$

where s is the step length in meter, f is the time duration of a step in second, and a, b are person-dependent constants. In SpyLoc client application, we want to leverage the ranging scheme and the step detection algorithm to build the above model for each person with no or minimal efforts.

- We like to utilize the sensor fusion technique to collect more cleaner rotation/direction sensor reading. However, despite the surrounding noise, these sensor also possess some biased noise. We like to address this challenge by following some pre-defined steps to calibrate the rotation sensor reading.

- We observe that the range detection from the NLoS *beacon device* shows less accuracy compare to the LoS *beacon device*. Therefore, we want to address the challenge of differentiating the LoS *beacon device* from the NLoS *beacon device* before estimating the range.

- In range estimation, knowing the temperature of the environment is important to calculate the speed of the sound. In our best knowledge, smartphones usually don't have the temperature sensor to provide the temperature of the surrounding environment. However, we want to utilizes the audio interface of the smartphone and the propagation speed of the acoustic signal to estimate the environment temperature.